WeatherNet Weather Forecast Using ML Techniques

Dor Shabat and Yuval Rozner

BRAUDE College of Engineering, Karmiel

Capstone Project Phase A - 61998

Dr. Dan Lemberg and Mrs. Elena Kramer

September 21, 2024

Contents

Abstract

WeatherNet presents an innovative system for forecasting weather in Israel using advanced machine learning techniques. Focusing on mid-term weather predictions, specifically temperature forecasts for Israel's northern, central, and southern regions, the system leverages historical and real-time data from the Israel Meteorological Service (IMS) to develop a machine learning-based solution.

At the core of this system will be a hybrid architecture, combining 1D Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTM), and attention mechanisms. This approach will capture short-term temporal patterns, long-term dependencies, and spatial relationships between weather stations. The system will provide reliable forecasts up to four days in advance, aiming to model complex weather phenomena and deliver accurate predictions across Israel’s regions.

The system will be accessible through a web-based platform, which will provide real-time weather forecasts. The platform will offer interactive visualizations and insights into the prediction accuracy and underlying architecture, allowing users to explore and understand the machine learning methods driving the forecasts. WeatherNet introduces a machine learning-driven approach, broadening the range of techniques available for accurate weather forecasting in Israel.

Keywords: *Time Series Forecasting, Israeli Weather Forecasting, Machine Learning (ML), Hybrid Architecture, Web-Based Platform.*

Introduction

Weather forecasting is a complex task, requiring the analysis of various atmospheric parameters, such as temperature, humidity, wind speed, and pressure. Traditionally, numerical weather prediction (NWP) models have been the go-to method for forecasting. These models use mathematical equations based on physical laws to simulate the atmospheric processes driving weather changes. While effective in many cases, traditional NWP models face limitations, primarily when tasked with handling non-linear relationships between weather parameters and processing vast amounts of data efficiently.

Machine learning models like LSTMs and CNNs offer significant advantages in weather forecasting by addressing the limitations of traditional NWP models, particularly in handling large, complex datasets. These models excel at identifying non-linear patterns between atmospheric variables and are well-suited for processing vast amounts of historical weather data (Casolaro et al., 2023; Bochenek & Ustrnul, 2022). This capability enables more accurate predictions of various weather parameters, such as precipitation and temperature, by capturing relationships between factors like wind, humidity, and pressure (Ahmadi et al., 2014). For example, LSTMs have been shown to significantly improve rainfall prediction accuracy in localized regions, while CNNs have demonstrated high accuracy, up to 97%, in temperature forecasting, further proving the robustness of ML models in handling diverse weather conditions (Bochenek & Ustrnul, 2022).

In WeatherNet, we aim to utilize these advantages of advanced ML techniques to develop a new mid-term weather forecasting solution for Israel. Our system will focus on predicting weather conditions, primarily temperature, for three key regions: the northern, central, and southern parts of the country. By leveraging a hybrid architecture that integrates CNNs, LSTMs, and attention mechanisms, we aim to capture both short-term and long-term temporal patterns, as well as spatial relationships between weather stations. The system will be deployed as a web-based platform, making real-time weather predictions accessible to users, with interactive tools to explore forecast accuracy and the underlying machine learning architecture.

Literature Review

Weather

Weather refers to the dynamic atmospheric conditions at a given location, influenced by factors such as temperature, humidity, wind speed, and atmospheric pressure, all of which interact to create the varying weather patterns we experience daily. These elements, along with the Earth's rotation and energy from the sun, continuously interact with land, oceans, and the atmosphere, causing fluctuations that we perceive as changes in weather. These shifts can occur rapidly, influencing everything from a clear sky turning cloudy to the sudden onset of rain or wind. Understanding weather's variability is essential not only because it directly affects our daily activities, but also because it governs critical natural processes on a global scale.

The significance of weather goes beyond our choice of clothing or whether we carry an umbrella - it influences major sectors of human activity. Agriculture, for instance, depends heavily on weather conditions to ensure optimal planting and harvesting times. Renewable energy sources such as wind and solar power are also directly affected by atmospheric conditions. Furthermore, infrastructure planning must take weather into account to ensure that buildings, roads, and other critical systems can withstand extreme conditions like floods, storms, or prolonged heat. Accurate weather predictions play a crucial role in managing these risks, offering the potential to save lives, protect property, and enhance operational efficiency across industries. (Weather, 2023, *Wikipedia*)

Timeframes of Weather Prediction

Weather forecasts are categorized into short-term, mid-term, and long-term predictions, each serving different purposes. ***Short-term*** forecasts, covering the next 6 hours, help with immediate decisions like outdoor events or flight operations. ***Mid-term*** forecasts span several days to a week, assisting with planning activities such as travel or farming. ***Long-term*** forecasts extend months ahead, focusing on seasonal trends useful for agriculture and energy management.

WeatherNet focuses on mid-term forecasting, aiming to provide insights into key weather parameters over the coming days, allowing users to prepare for weather changes during that period.

Israel Climate and Weather

Israel's climate is predominantly Mediterranean, characterized by hot, dry summers and cool, rainy winters. The country experiences significant regional variations, with coastal areas being more humid, desert regions like the Negev being arid, and the northern areas receiving more rainfall, especially during winter.

IMS - Israel Meteorological Service

Ministry of Transport and Road Safety
IMS Israel Meteorological Service

figure 01. Israel Meteorological Service.
Note. Source: [Israel Meteorological Service, n.d.]

The *Israel Meteorological Service (IMS)* is a critical organization responsible for providing accurate and reliable meteorological data to safeguard lives and property. Operating under the *Ministry of Transport and Road Safety*, *IMS* plays a vital role in supporting the country's economic and social development by offering services such as weather forecasting, climate analysis, and warnings for severe weather events. (Israel Meteorological Service, n.d.).

IMS's Measurement Stations

The IMS operates a comprehensive network of measurement stations across the country to gather real-time weather data. This network includes 82 automated stations that are strategically distributed from the northern regions to the southern desert areas. These stations continuously monitor key weather parameters such as temperature, humidity, wind speed, and precipitation, enabling accurate and timely weather forecasting. (Israel Meteorological Service, n.d.)

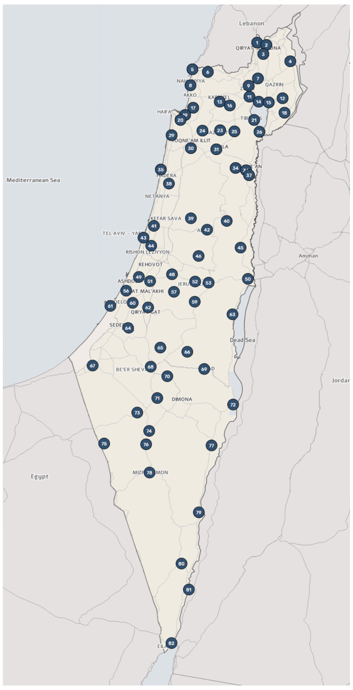


figure 02. IMS Stations On Map.
Note. Source: [Israel Meteorological Service, n.d.]

תאור	יחידות מדידה	משתנה
לחץ בגובה התחנה	hPa	BP
קרינה מפוזרת	w/m^2	DiffR
קרינה גלובלית	w/m^2	Grad
קרינה ישירה	w/m^2	NIP
כמות גשם	mm	Rain
לחות יחסית	%	RH
סטיית התקן של כיוון הרוח	deg	STDwd
טמפרטורה יבשה	°C	TD
טמפרטורה מקסימלית	°C	TDmax
טמפרטורה מינימלית	°C	TDmin
טמפרטורה ליד הקרקע	°C	TG
זמן סיום 10 הדקות המקסימליות	hhmm	Time
כיוון הרוח	deg	WD
כיוון המשב העליון	deg	WDmax
מהירות הרוח	m/s	WS
מהירות רוח 10 דקתית מקסימלית	m/s	Ws10mm
מהירות רוח דקתית מקסימלית	m/s	WS1mm
מהירות המשב העליון	m/s	WSmax
גשם בדקה	mm	Rain_1_min

figure 03. IMS Recorded Parameters.
Note. Source: [Israel Meteorological Service, n.d.]

Measured Weather Parameters

Refer to *Figure 3* for an overview of the key weather parameters recorded by the IMS, with further details provided in the following sections.

- **Dry Temperature**:

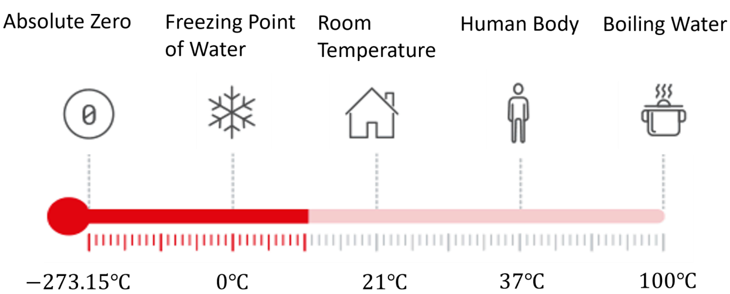


figure 04. Temperature in Celsius Degrees.

- Unit: *°C*
- Explanation: The dry temperature is the air temperature measured by a standard thermometer without the influence of moisture. It is one of the most fundamental

weather parameters and is essential in forecasting, as it directly impacts other variables like humidity, wind, and precipitation.
The unit, degrees Celsius *(°C)*, is used universally, with 0*(°C)* defined as the freezing point of water and 100*(°C)* as its boiling point, making it crucial for assessing temperature fluctuations and weather patterns. (Temperature, 2023, *Wikipedia*)

- <u>Measured Variables</u>:
    - ◆ **TD:** The average temperature measured during the specified time period *(10 minutes)*.
    - ◆ **TDmax:** The highest temperature recorded during the time period.
    - ◆ **TDmin:** The lowest temperature recorded during the time period.
    - ◆ **TG:** The temperature measured at grass level, which helps assess surface-level cooling and frost conditions.

o **Barometric Pressure**:

- <u>Unit</u>: *hPa*
- <u>Explanation:</u> Atmospheric pressure at the height of the station, measured in hectopascals *(hPa)*. Barometric pressure changes significantly with altitude, as pressure decreases with increasing height. Monitoring pressure at specific altitudes is crucial for accurate forecasting. Pressure variations are key indicators of shifting weather patterns, with drops signaling the approach of storms, while rising pressure suggests improving conditions.
The unit hectopascal *(hPa)* is commonly used in meteorology, where 1 *hPa* is equivalent to 100 *pascals*, a standard unit for measuring force per area.
- <u>Measured Variable</u>:
    - ◆ **BP:** Measured in hectopascals *(hPa)* during the specified time period. This variable is crucial for understanding atmospheric pressure trends and predicting weather changes.

o **Humidity**:

- <u>Unit</u>: *percentage (%)*

- Explanation: Humidity refers to the amount of water vapor present in the air. There are different types of humidity measurements. The most commonly used in weather forecasting is relative humidity (RH).
  Relative humidity expresses the percentage of moisture in the air compared to the maximum amount the air can hold at a specific temperature. For example, an RH of 50% means the air is holding half of the water vapor it could potentially hold at that temperature. High RH indicates more moisture in the air, which can lead to cloud formation and precipitation, while low RH suggests drier conditions. When RH reaches 100%, the air is fully saturated, leading to condensation and possible fog, dew, or precipitation.
  Absolute humidity, which represents the actual amount of water vapor in the air, can be calculated using the dry temperature and relative humidity through formulas that determine vapor pressure.

- Measured Variable:

  - **RH**: Measured as a percentage *(%)* during the time period, it provides key insights into moisture levels and the likelihood of precipitation or fog.

- **Radiation**:

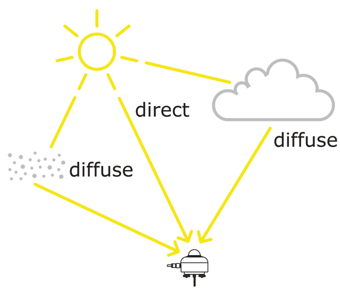


figure 05. *Solar Radiation.*

- Unit: *W/m²*

- Explanation: Radiation refers to the energy emitted by the Sun that reaches the Earth's surface. It plays a crucial role in shaping weather conditions by influencing temperature, cloud formation, and atmospheric circulation. Solar radiation is measured in watts per square meter *(W/m²)*, representing the amount of solar energy received on a surface over time. This unit is essential for quantifying the Sun's energy and understanding its impact on weather patterns and solar energy potential. The IMS records three types of solar radiation - direct, diffuse, and global radiation.

- Measured Variable:

    - **NIP (Direct Radiation)**: Measures solar radiation that reaches the Earth's surface directly from the Sun without being scattered by the atmosphere. It is critical for assessing solar energy potential on clear days and is used to determine the intensity of sunlight for solar panels. Direct radiation also influences surface heating and weather phenomena such as heat waves.

    - **Diff (Diffuse Radiation)**: Measures solar radiation that has been scattered by molecules and particles in the atmosphere before reaching the surface. This is essential for understanding the solar energy available even under cloudy or overcast conditions, as it accounts for the indirect sunlight that still reaches the ground.

    - **Grad (Global Radiation)**: Represents the total solar radiation reaching the surface, combining both direct and diffuse radiation. Global Radiation is approximately the sum of direct and diffuse radiation, adjusted by the solar zenith angle (θ), which accounts for the Sun's position relative to the surface. This parameter is key for understanding energy balance and solar potential, with the relationship expressed as:

    $$\text{Global Radiation} = \text{Direct Radiation} \times \cos(\theta) + \text{Diffuse Radiation}$$

o **Wind**:

- Unit: speed in *m/s*, direction in *degrees*

- Explanation: Wind is the natural movement of air or other gasses relative to a planet's surface, typically moving from areas of high pressure to low pressure. Wind speed and direction are critical for understanding weather dynamics and the movement of weather systems. By analyzing wind direction, it is possible to predict how air masses and storms will move across regions. Wind is a crucial parameter in weather forecasting, as it influences temperature distribution, storm development, and precipitation patterns. (Wind, 2023, *Wikipedia*).

- Measured Variable:
    - **WD (Wind Direction)**: The direction from which the wind is blowing, measured in degrees. It indicates the source of the wind (e.g., 0° for north wind, 90° for east wind).
    - **WS (Wind Speed)**: The speed of wind measured at the station's height, expressed in meters per second (m/s).
    - **WDmax (Gust Wind Direction)**: The direction of the strongest wind gusts, providing insight into short-term, high-velocity wind shifts.
    - **WSmax (Gust Wind Speed)**: The speed of the strongest wind gusts recorded during a time period, typically higher than sustained wind speeds.
    - **STDwd (Standard Deviation of Wind Direction)**: Measures the variability or fluctuation in wind direction over a given time period.

o **Rain**:

- Unit: *millimeters (mm)*
- Explanation: Rain refers to the total amount of precipitation that falls over a specific area. It is one of the key factors in weather forecasting and water resource management. Measuring rainfall helps in predicting potential flooding, assessing drought conditions, and supporting agricultural planning. The intensity and duration of rainfall can also influence temperature and humidity levels, impacting broader weather patterns.
- Measured Variable:
    - **Rain:** The total amount of rainfall accumulated over a time period.

Traditional Weather Forecasting

Traditionally, weather forecasting is based on *numerical weather prediction (NWP)* models, which rely on the principles of physics and complex mathematical equations to simulate atmospheric behavior. This task is often a collaborative effort among multiple countries. Organizations like the *European Centre for Medium-Range Weather Forecasts (ECMWF)*, *American Meteorological Service*, *German Meteorological Service (DWD)*, and research institutes like the *Max Planck Institute* all contribute to advancing global weather prediction through data sharing and model development, ensuring more accurate and reliable forecasts worldwide.

There are three primary stages in the weather forecasting process:

1. Observing current weather conditions: This involves collecting data from satellites, radars, weather stations, and weather balloons to provide a clear picture of the atmosphere's current state.

2. Predicting future changes: *NWP* models take this data and simulate future weather by applying physical and mathematical principles to forecast how atmospheric conditions will evolve.

3. Refining the forecast: Meteorologists review model outputs, applying their expertise to adjust for local nuances, historical patterns, and known model biases, improving the precision of the forecast.

In addition to these steps, *ensemble forecasting* is increasingly used to manage uncertainty. By running multiple simulations with slightly varied initial conditions, *ensemble forecasting* offers a range of potential outcomes, helping forecasters assess probabilities and improve reliability, especially for mid and long-range forecasts. Techniques such as *data assimilation*, which incorporates real-time observational data into models, and *nowcasting*, used for very short-term predictions, are also key elements in modern weather forecasting.

As mentioned earlier, our system focuses specifically on mid-range weather forecasting, predicting conditions over several days. Next, we will explore the primary models used by the *IMS* for this purpose (Israel Meteorological Service, n.d.).

Integrated Forecasting System (IFS) - of ECMWF

The *European Centre for Medium-Range Weather Forecasts (ECMWF)* uses the *Integrated Forecasting System (IFS)* to produce global weather forecasts with a 9 km resolution, running four times daily to predict up to 240 hours ahead. *IFS* supports regional models like *COSMO* and *ICON*. It integrates real-time data through an advanced assimilation system and includes components like oceans. *IFS* has two modes: *HRES* for high-resolution forecasts and *ENS* for probabilistic ensemble forecasts, offering predictions up to 15 days ahead (European Centre for Medium-Range Weather Forecasts, n.d.).

The Global Forecast System (GFS) - of the American meteorological service

The *Global Forecast System (GFS)* from the *National Centers for Environmental Prediction (NCEP)* simulates atmospheric conditions globally using the *Finite Volume Cubed-Sphere (FV3)* model with a 13 km resolution. Running four times a day, *GFS* forecasts up to 16 days ahead, providing hourly predictions for the first five days and 3-hourly ones after that. The *Global Data Assimilation System (GDAS)* processes observational data to initialize the model (National Centers for Environmental Prediction, n.d.).

United Kingdom MET Office (UKMO) - of the UK's national weather service

The *UKMO Model*, developed by the UK's *Met Office*, is part of the *Unified Model (UM)*, which forecasts both weather and climate across different timescales. It offers global forecasts with a 10 km resolution and is also used for high-resolution regional predictions. The model operates in both deterministic and ensemble modes to handle uncertainty in predictions (Met Office, n.d.).

ICON Model - of the German meteorological service

The *ICON Model*, developed by the *German Meteorological Service (DWD)* and *Max Planck Institute*, predicts weather and climate using icosahedral grids for global and regional forecasts. It integrates atmospheric, oceanic, and land data and runs efficiently on supercomputers. *ICON* supports both weather forecasting and climate simulations and is available as open-source software, promoting collaboration (Max Planck Institute for Meteorology, n.d.).

COSMO Model (COnsortium for Small-scale Modeling)

The *COSMO Model (COnsortium for Small-scale Modeling)* is a high-resolution weather prediction model, including *IMS* as a participant. It specializes in regional forecasts with a resolution as fine as 2.8 km, capturing small-scale phenomena like thunderstorms. *COSMO* is used for short- and medium-term forecasts in regions needing precise weather predictions, such as Israel (Israel Meteorological Service, n.d.).

Time Series

A time series is a sequence of data points collected or recorded at successive, equally spaced points in time. Unlike traditional datasets that assume data points are independent, time series data captures the evolution of values over time, making temporal ordering a crucial factor in its analysis. Time series analysis focuses on studying patterns within the data to understand past behavior and predict future trends.

Time series data is used in a wide range of fields, including economics, finance, biology, and meteorology. In finance, for example, stock prices are recorded over time and analyzed to forecast future market behavior. In biology, time series can represent the growth of populations or the spread of diseases. Regardless of the field, the goal of time series analysis is typically to recognize underlying patterns, trends, and cyclic behaviors to make informed decisions or predictions.

Key Characteristics of Time Series:

- **Temporal Dependency:** In time series, data points are connected, meaning each point is influenced by previous and future points, introducing challenges not seen in independent data.
- **Trend:** Long-term movement in the data, indicating a general direction over time, such as economic growth or decline.
- **Seasonality:** Recurring patterns at regular intervals, like seasonal effects in sales or temperature fluctuations.
- **Noise:** Random variations that are not part of trends or seasonality, often needing filtering to reveal underlying patterns.
- **Stationarity:** A series is stationary if its statistical properties, like mean and variance, are constant over time. Many models assume stationarity, so non-stationary data often needs to be transformed.

Time Windowing in Time Series Analysis

Time windowing is a crucial concept in time series analysis that involves framing a sequence of historical data points to predict future outcomes. The idea is to take a "window" of consecutive observations from the past and use them as inputs to predict one or more future values. This technique is widely used in forecasting models to capture both short-term and long-term temporal patterns.

Sliding Windows Approach

One common method in time windowing is the sliding window approach. This involves shifting the window over time to generate multiple input-output pairs from a single time series dataset. For example, given a time series of weather data, a sliding window might take the first 24 hours as the input window to predict the 25th hour. The window then shifts one step forward to predict the 26th hour, and so on. This method ensures that the model is trained on a wide variety of input-output pairs, improving its ability to generalize and make accurate predictions.

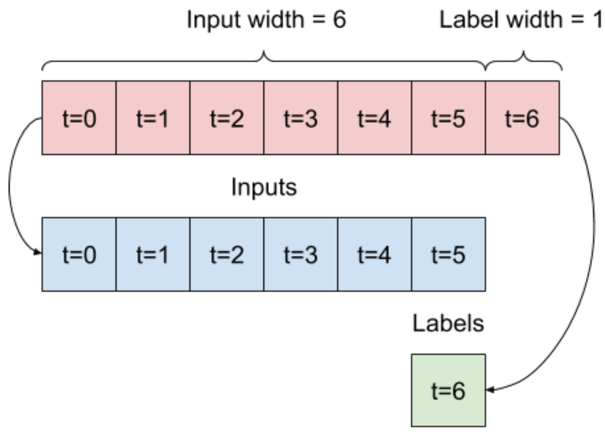


figure 06. Time Window Example.
Note. This figure shows a simple time window approach with 6 time steps as input (t=0 to t=5) and one time step (t=6) as the prediction target. Source: [TensorFlow].

Optimal Window Sizes

The size of the time window is a key hyperparameter that can significantly affect model performance. If the window size is too small, the model might miss long-term trends or dependencies. On the other hand, if the window size is too large, the model may become overwhelmed with unnecessary or redundant information, making it harder to learn meaningful patterns.

Choosing the right window size depends on the specific problem being solved, the characteristics of the data, and the forecast horizon. In many cases, experimentation is needed to find the optimal window size for a given application.

Machine Learning

Convolutional Neural Networks

CNNs are a type of deep learning model designed to process grid-like data, commonly used in image recognition. CNNs use convolutional layers to extract local features by sliding filters over the input, followed by pooling layers to reduce the dimensionality and focus on the most important information.

1D CNNs

1D Convolutional Neural Networks (1D CNNs) are specifically tailored for sequential data. Instead of looking for spatial features, they detect temporal patterns by applying convolution across one dimension, typically the time axis. This makes them effective for tasks like signal processing and time-series forecasting, where local temporal patterns are crucial.

Recurrent Neural Networks - LSTM

RNNs are designed to handle sequential data by maintaining a hidden state that carries information from previous time steps. RNNs excel at learning temporal dependencies, making them suitable for time-series forecasting. However, traditional RNNs struggle with long-term dependencies, which is addressed by more advanced versions like Long Short-Term Memory (LSTM) networks, which manage memory more effectively over time.

Attention Mechanism

The attention mechanism allows neural networks to focus on the most important parts of the input by assigning different weights to each element. This helps the model prioritize relevant information, improving its ability to identify key patterns and relationships across the input data. Unlike traditional models, attention processes all input data simultaneously, capturing complex relationships in the sequence. It is a key feature in transformer models, enabling them to efficiently process and analyze data.

ResNet

ResNet, or Residual Networks, introduced the concept of skip connections, allowing data to bypass certain layers in deep networks. Skip connections preserve important information by ensuring key features are retained throughout the network.

Forecasting Approaches

In time series forecasting, there are two primary strategies for predicting multiple time steps: Recursive Forecasting and One-Shot (Direct Multi-Step) Forecasting. Each approach offers distinct advantages depending on the forecast horizon and the desired precision.

Recursive Forecasting (Autoregressive Models)

This strategy involves predicting one time step ahead at a time and then using that prediction as input for the next time step. The process is recursive because each new prediction depends on the prior forecast, creating an autoregressive loop. The main advantage of recursive forecasting lies in the fact that it focuses on predicting one time frame into the future at each step. This allows the model to be designed in such a way that it can optimize its predictions for each individual time step, leading to high accuracy in short-term forecasts.

However, one of the key challenges of recursive forecasting is error propagation. As each prediction depends on the previous one, any small error in predicting an earlier time step can accumulate and magnify over time, degrading the quality of predictions for later steps. This makes recursive forecasting less reliable for longer-term predictions, as the errors compound over multiple steps.

One-Shot (Direct Multi-Step) Forecasting

One-shot forecasting predicts multiple time steps in a single pass. Instead of making one prediction at a time, the model outputs the entire sequence of future predictions at once.

This approach completely avoids the issue of error propagation, as all time steps are predicted simultaneously without depending on previous predictions. It is particularly advantageous for mid to long-term forecasting, where capturing broader trends over an extended time horizon is more important than fine-grained precision at individual time steps. An additional benefit of one-shot forecasting is that it can be less computationally intensive, as the model only needs to make one forward pass to predict all future time steps, which can make the forecasting process more efficient.

When predicting longer periods, the model's weights must balance performance across the entire forecast window, but focusing on a broader time span can reduce accuracy for individual time steps.

Expected Achievements

System Goals

The primary goal of this project is to develop an Israeli weather forecasting website that showcases our real-time predictions alongside comprehensive weather statistics. The website will serve as both an informative platform for general users and a detailed resource for evaluating the accuracy of our forecasting models.

At its core, the website will rely on our advanced machine learning models to deliver accurate, real-time weather forecasts for three key regions of Israel: Haifa, Tel-Aviv, and Beer-Sheva. The backend will be powered by our forecasting system, which will automatically import real-time data from the *Israel Meteorological Service (IMS)* API. This data will be processed by our models to generate reliable predictions for the upcoming days.

In addition to providing real-time forecasts, the website will include:

- **Ongoing Weather:** Displaying real-time weather parameters, including temperature, humidity, wind speed, and more.
- **Prediction Accuracy Insights:** The site will track the precision of our forecasts by comparing predictions from previous days with actual recorded data from *IMS*. Users will be able to visually explore how closely our predictions align with reality.
- **Interactive Comparison Tool:** A feature that allows users to compare our model's forecasts with traditional IMS predictions, offering transparency in the accuracy of machine learning-based forecasting versus traditional methods.
- **Research and Development Section:** Users will also have the option to read about and explore our project papers and the entire research process, including the architecture we developed, the implementation of the models, and other technical insights. This will allow visitors to gain a deeper understanding of the underlying technology and methodologies driving our system.

The website will be designed with a focus on user experience, ensuring it is easy to navigate, visually engaging, and informative. Users will not only be able to view predictions but also investigate detailed statistics about the precision of our forecasts, enhancing understanding of our machine learning model's performance.

Forecast Precision Expectations

Regions of Focus:

WeatherNet System will focus on weather predictions in three key regions of Israel:

- Haifa (north), Tel-Aviv (center), Beer-Sheva (south)

Prediction Focus:

Our main goal is to accurately forecast **temperature parameters** in these regions by leveraging:

- Historical weather data
- Real-time recordings from numerous meteorological stations across Israel

Forecast Duration:

We aim to provide reliable forecasts for **at least four days ahead**.

Accuracy Target:

Our success criteria are defined by:

- An error margin of no more than **1.5°C**
- Comparison with traditional forecasts from the **Israel Meteorological Service (IMS).** Matching or exceeding the precision of IMS forecasts would be considered a significant achievement

Forecasting Approach:

To accomplish these goals, we will utilize an **ensemble forecasting approach**:

- **Model 1**: Provides hourly predictions for the next 24 hours using a **recursive method.**
- **Model 2**: Generates predictions for subsequent days using a **one-shot approach** with flexible time steps.

By combining these two models, we aim to:

- Enhance forecast accuracy.
- Extend the reliable forecast period.

Engineering Process

Process

Progress and Results So Far

Connection with IMS and Data Access

Early in the project, we contacted the *Israel Meteorological Service (IMS)* through emails to request access to their weather data and inquire about their forecasting methods. Through this interaction, we successfully gained access to the *IMS* API, which allows us to retrieve real-time and historical weather data. Additionally, the *IMS* offered to provide us with their past forecast data, which will enable a direct comparison between their *Numerical Weather Prediction (NWP)* models and our machine learning-based forecasts. This connection with the *IMS* has been invaluable for ensuring that we have the necessary data resources to develop and validate our models.

Preliminary Research

We conducted extensive research on key aspects of weather forecasting to lay the groundwork for our project. This included understanding weather parameters like temperature, humidity, and wind, and exploring the differences between short-, mid-, and long-term weather predictions. We also studied Israel's climate and the role of the *IMS* in forecasting. A major focus was on comparing traditional Numerical Weather Prediction *(NWP)* models with machine learning techniques such as *CNNs* and *LSTMs*, identifying how machine learning can overcome the limitations of traditional methods. Additionally, we explored time series analysis, sliding windows, and forecasting approaches, including recursive and one-shot methods, to determine the best techniques for our system.
This research guided the design of our models and the overall system architecture.

Exploratory Data Analysis (EDA)

We began data exploration by using the *IMS* API to retrieve past weather data from a single station. This allowed us to familiarize ourselves with the structure and characteristics of the data. In this phase, we addressed data preprocessing challenges, including handling missing values, normalizing variables, and transforming cyclic parameters like time and seasonal effects. We also experimented with various approaches for visualizing the data to better understand trends and anomalies in weather parameters. This initial EDA helped us define preprocessing workflows and set the foundation for more advanced modeling.

Exploration and Implementation of Simple Models

During this phase, we experimented with several simple models to gain a deeper understanding of the data and forecasting approaches. These included basic models like linear regression, persistence models, and early implementations of machine learning techniques like *Convolutional Neural Networks (CNNs)* and *Recurrent Neural Networks (RNNs)*. The aim was to test how these basic approaches handled weather parameter forecasting and to identify areas for improvement. Each of these models provided a baseline for performance, helping us to evaluate the complexity of weather forecasting tasks. This exploration allowed us to identify strengths and weaknesses in simple models and guided us toward developing more complex, tailored architecture.

Development of a Novel Architecture

We designed a hybrid architecture combining *Convolutional Neural Networks (CNNs)*, *Long Short-Term Memory (LSTM)* networks, and *attention mechanisms* to capture both short-term and long-term weather patterns. In designing this architecture, we considered the advantages and disadvantages of each component, as discovered during the preliminary research and the implementation of simple models. This architecture, which supports both *recursive* and *one-shot* forecasting, is tailored to improve prediction accuracy. A detailed description of the architecture is provided later in the paper.

Set Evaluation Criteria

To assess the performance of our forecasting models, we defined key evaluation metrics, including *Mean Absolute Error (MAE)* and *Root Mean Squared Error (RMSE)*. These metrics will help us measure both the average error and the impact of larger deviations in predictions. The evaluation criteria will be essential in comparing our machine learning models against the traditional *IMS* forecasts.

Defined Product Vision and Objectives

We established the core goals for our system, which will focus on providing accurate, real-time weather forecasts for Israel’s northern, central, and southern regions. The system will be accessible via a web-based platform, featuring a user-friendly interface with interactive visualizations, forecast accuracy insights, and comparison tools. To support this, we created a UML *activity diagram* of the system's operational flow and drafted diagrams of the website screens, outlining the user interface and experience. This design ensures that both general users and technical experts can engage with and benefit from the forecasts.

Writing the Paper

Throughout the development process, we documented our research, progress, and findings in a formal academic paper. This paper includes detailed sections on the project's goals, architecture, research methodologies, and technical achievements. It serves as both a comprehensive report of our work so far and a guiding document for future development phases.

Preliminary Findings and Insights

In this section, we present the findings from our exploratory analysis using data from the Jerusalem weather station, spanning the years 2000 to 2023. Our focus was on experimenting with various techniques for data manipulation and implementing simple forecasting models for temperature prediction, utilizing basic time windowing approaches.

The following are some key achievements, including visualizations and results from these initial experiments.

	count	mean	std	min	25%	50%	75%	max
BP (hPa)	160487.0	922.490836	3.683396	907.04	919.76	922.3	924.93	937.44
RH (%)	160231.0	57.103195	24.117456	4.00	37.00	56.0	79.00	100.00
TD (degC)	209755.0	17.955506	7.041178	-1.70	12.10	18.3	23.30	41.60
TDmax (degC)	209760.0	18.084489	7.084270	-1.60	12.30	18.4	23.40	42.00
TDmin (degC)	209757.0	17.829548	6.999501	-1.80	12.00	18.2	23.10	41.40
WD (deg)	209910.0	236.100962	90.439883	0.00	180.00	273.0	300.00	360.00
WDmax (deg)	209856.0	235.435832	92.249904	0.00	173.00	273.0	301.00	360.00
WS (m/s)	209652.0	3.336034	1.635589	0.00	2.10	3.2	4.30	15.60
Ws1mm (m/s)	209647.0	4.489605	2.079202	0.00	3.00	4.3	5.70	21.70
Ws10mm (m/s)	209655.0	3.590693	1.685942	0.00	2.40	3.4	4.60	17.30
WSmax (m/s)	209654.0	5.719869	2.773302	0.00	3.70	5.4	7.30	28.00
STDwd (deg)	209889.0	18.782293	9.497722	0.00	13.00	17.0	21.90	99.90

figure 07. Statistical summary of key weather parameters from the Jerusalem station dataset.
Note. This summary provides an overview of the distribution and variability of these weather variables.

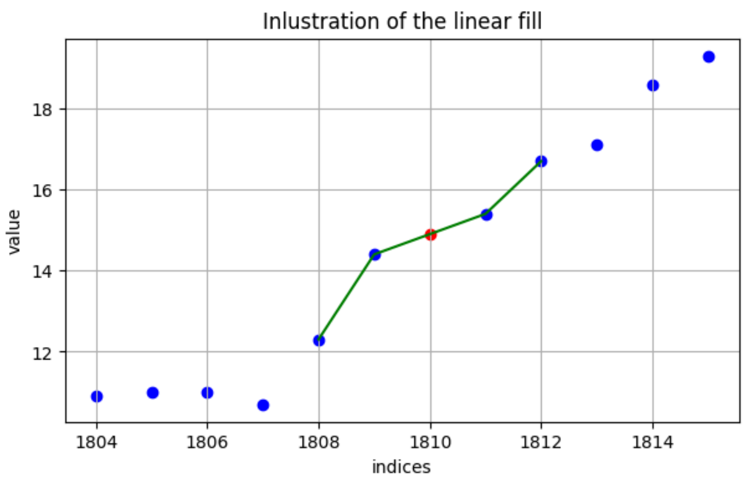


figure 08. Illustration of linear interpolation used for filling a single missing value in the dataset.
Note. The blue points represent the actual data values, and the red point represents the interpolated value, calculated as the average of its neighboring points.

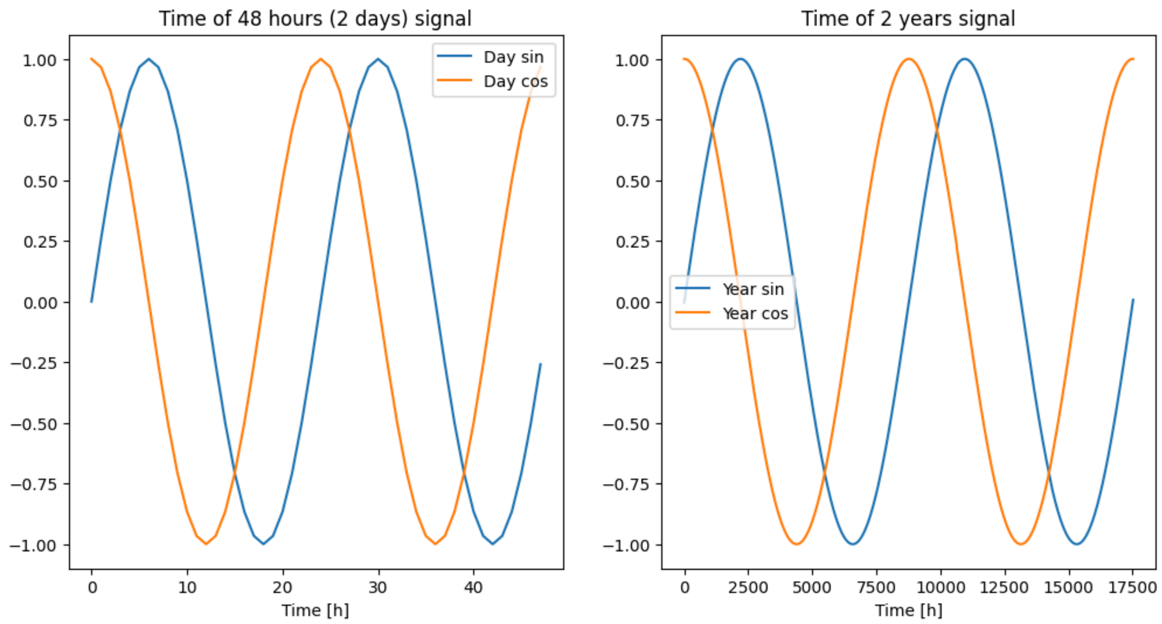


figure 09. Time Cyclic Parameters.
Note. The left plot shows the sinusoidal and cosinusoidal representations of daily cycles over 48 hours, while the right plot represents annual cycles over a two-year period. These transformations capture the cyclic nature of daily and yearly time series data, which are essential for modeling weather patterns with respect to time.

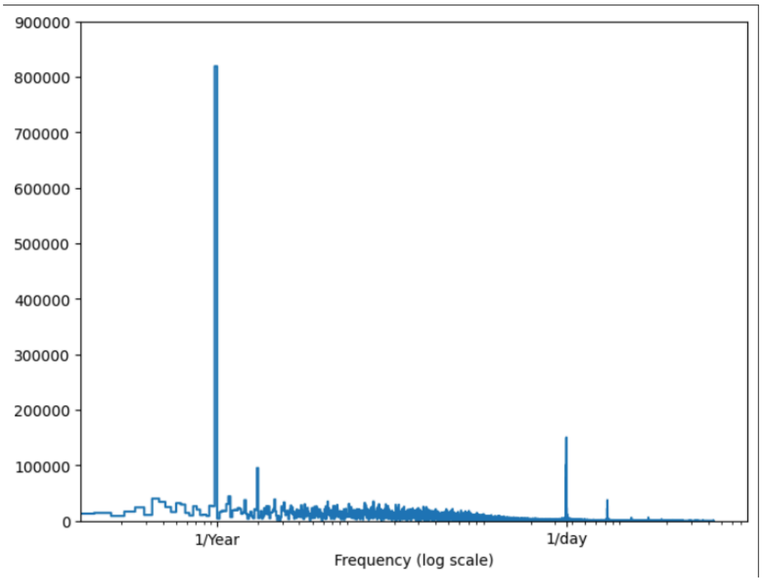


figure 10. Frequency Spectrum of Temperature Data Using Fast Fourier Transform (FFT).
Note. This figure illustrates the frequency spectrum of temperature data (in °C) using Fast Fourier Transform (FFT). The x-axis shows the frequency in a logarithmic scale, with key markers for 1/year and 1/day frequencies. The y-axis represents the magnitude of the Fourier components. The large peak around 1/year indicates strong annual periodicity in the temperature data, while smaller peaks closer to 1/day suggest daily cycles. This analysis helps in identifying dominant temporal patterns in the temperature fluctuations over the dataset. value, calculated as the average of its neighboring points.

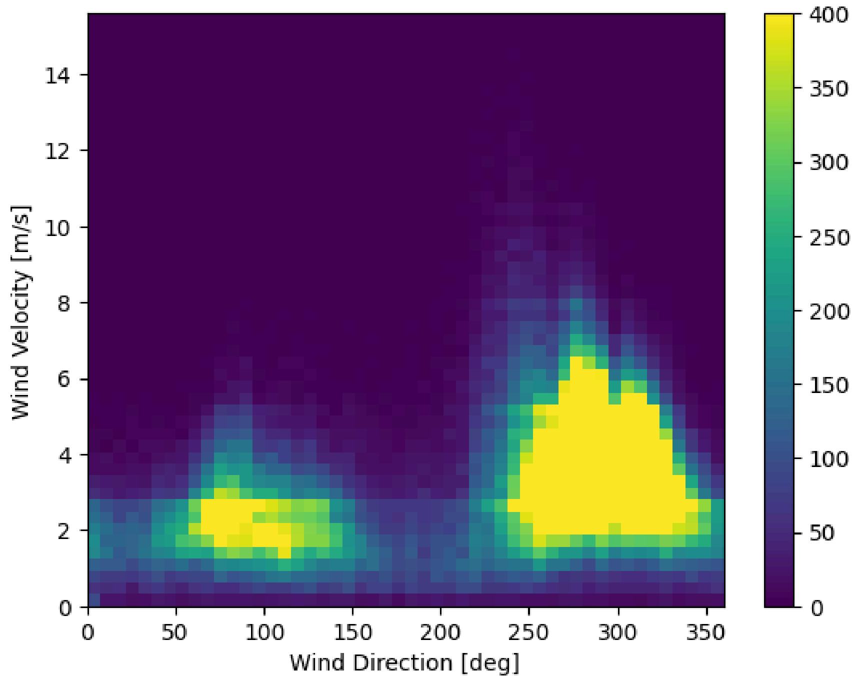


__figure 11.__ Heatmap showing the relationship between wind direction (0-360°) and wind velocity (0-15 m/s) at the Jerusalem station, before transforming wind direction and velocity into vector form.

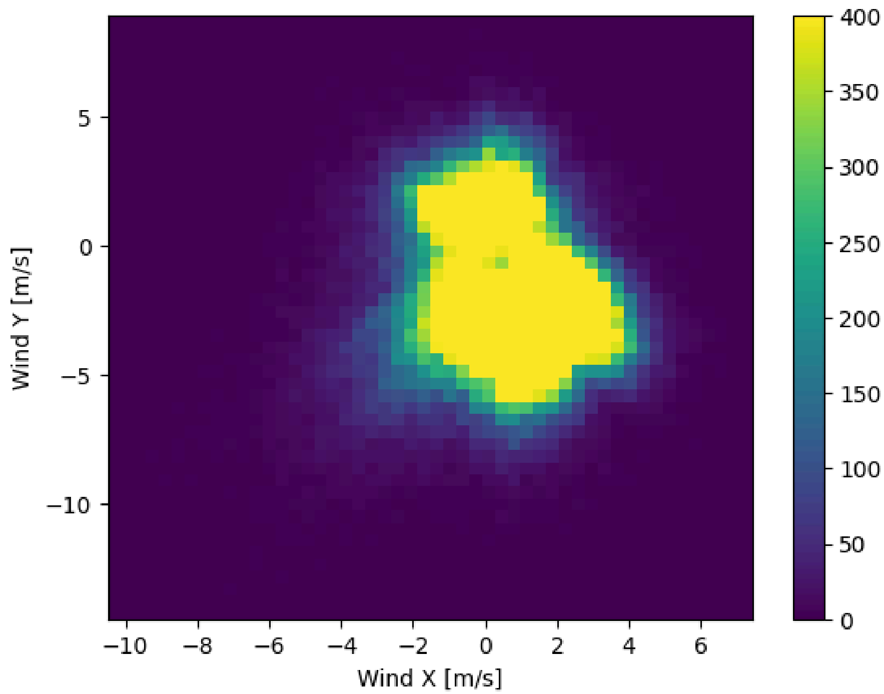


__figure 12.__ Heatmap showing the distribution of wind vectors after transforming wind direction and velocity into X (m/s) and Y (m/s) components at the Jerusalem station

	Date Time	BP (hPa)	RH (%)	TD (degC)	TDmax (degC)	TDmin (degC)	WD (deg)	WDmax (deg)	WS (m/s)	Ws1mm (m/s)	Ws10mm (m/s)	WSmax (m/s)	STDwd (deg)
0	01/01/2000 00:00	-	-	17.3	17.4	17.3	102	97	1.6	2.4	1.7	2.8	9.8
6	01/01/2000 01:00	-	-	17	17	16.9	79	58	2.1	2.9	2.2	3.6	14.2
12	01/01/2000 02:00	-	-	17.3	17.3	17.2	94	92	1.7	2.1	2.1	2.4	16.2
18	01/01/2000 03:00	-	-	16.7	16.8	16.7	81	75	1.5	1.8	1.9	2	8.6
24	01/01/2000 04:00	-	-	16.6	16.6	16.5	93	84	1.8	2	1.8	2.4	9.6

figure 13. *Sample of the first few rows of the dataset before modifications, displaying the original columns for date, time, and wind parameters.*

	BP (hPa)	RH (%)	TD (degC)	TDmax (degC)	TDmin (degC)	WDmax (deg)	Ws1mm (m/s)	Ws10mm (m/s)	STDwd (deg)	Wx	Wy	max Wx	max Wy	Day sin	Day cos	Year sin	Year cos
0	NaN	NaN	17.3	17.4	17.3	97.0	2.4	1.7	9.8	-0.332659	1.565036	-0.582153	2.738813	-5.461913e-12	1.000000	-0.004731	0.999989
6	NaN	NaN	17.0	17.0	16.9	58.0	2.9	2.2	14.2	0.400699	2.061417	0.686912	3.533858	2.588190e-01	0.965926	-0.004014	0.999992
12	NaN	NaN	17.3	17.3	17.2	92.0	2.1	2.1	16.2	-0.118586	1.695859	-0.167416	2.394154	5.000000e-01	0.866025	-0.003297	0.999995
18	NaN	NaN	16.7	16.8	16.7	75.0	1.8	1.9	8.6	0.234652	1.481533	0.312869	1.975377	7.071068e-01	0.707107	-0.002580	0.999997
24	NaN	NaN	16.6	16.6	16.5	84.0	2.0	1.8	9.6	-0.094205	1.797533	-0.125606	2.396711	8.660254e-01	0.500000	-0.001864	0.999998

figure 14. *Sample of the first few rows of the dataset after modifications, including additional columns for wind vector components and sinusoidal transformations for daily and yearly cycles.*

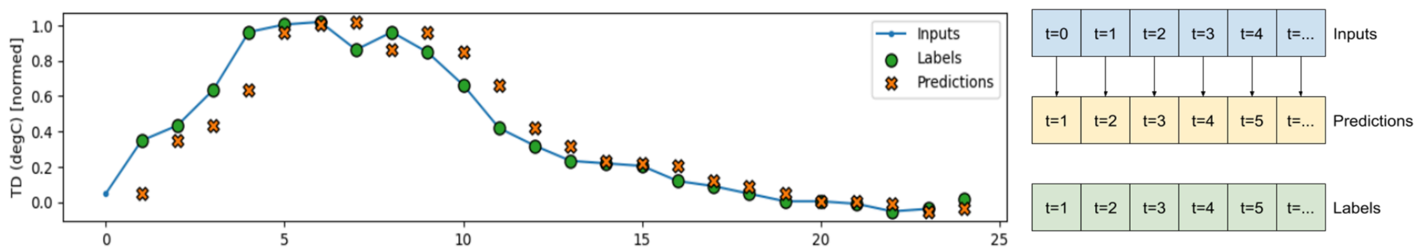


figure 15. Baseline Model Prediction.
Note. This figure shows the performance of a baseline model predicting temperature over time (in normalized °C).

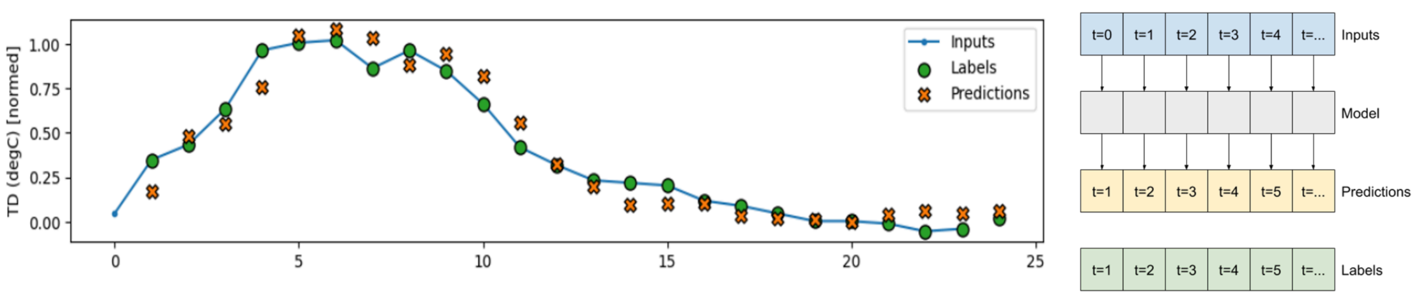


figure 16. Linear Model Prediction.
Note. This figure shows the performance of a linear model predicting temperature over time (in normalized °C).

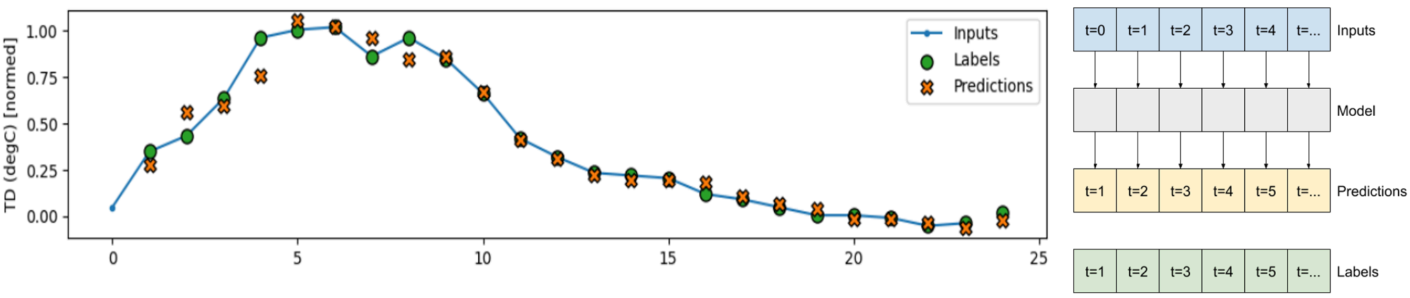


figure 17. Dense Model Prediction.
Note. This figure shows the performance of a dense model predicting temperature over time (in normalized °C).

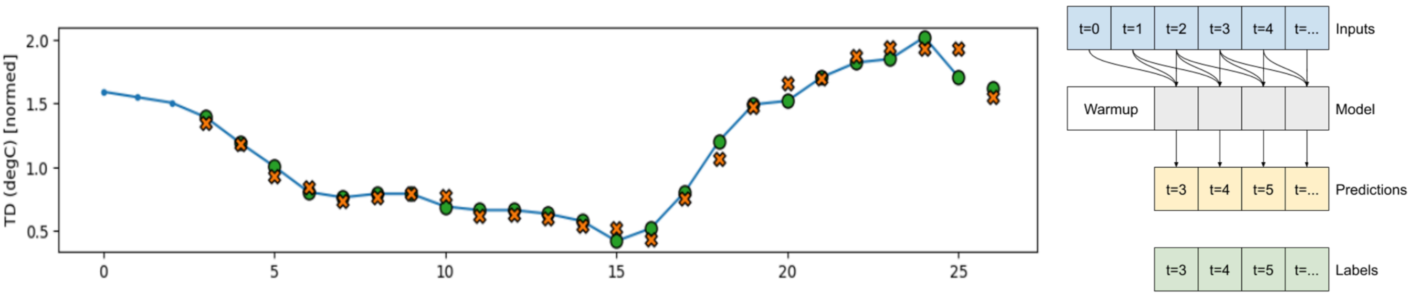


figure 18. CNN Model Prediction.
Note. This figure shows the performance of a convolutional neural network (CNN) model predicting temperature over time (in normalized °C).

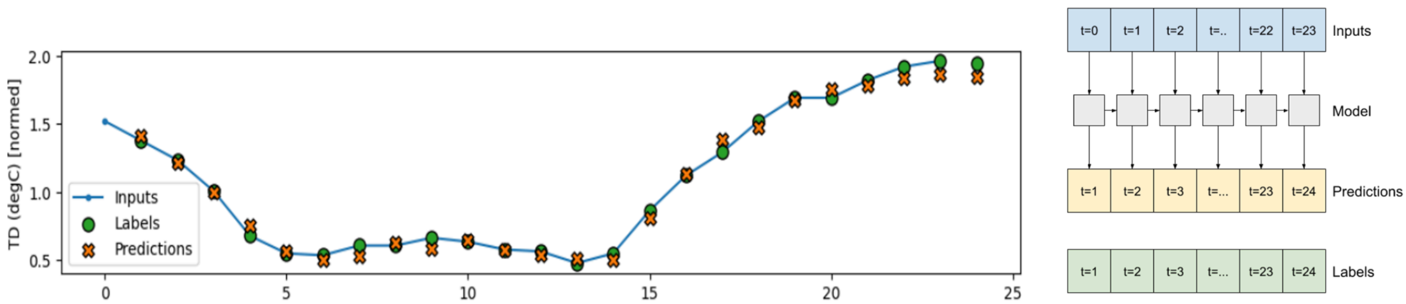


figure 19. LSTM Model Prediction.
Note. This figure shows the performance of an LSTM model predicting temperature over time (in normalized °C).

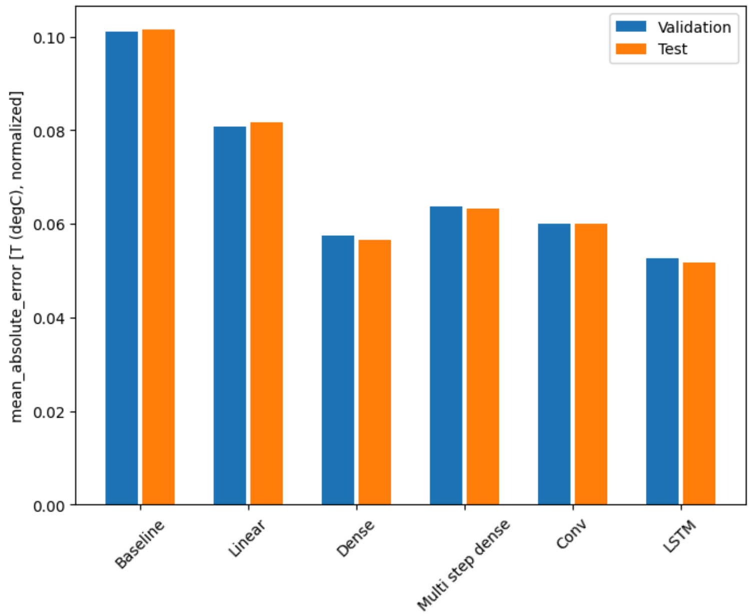


figure 20. Mean Absolute Error Comparison Across Models.
Note. This bar chart compares the mean absolute error (MAE) for different models (Baseline, Linear, Dense, Multi-step Dense, Conv, LSTM) across both validation and test datasets. The x-axis lists the models, while the y-axis represents the normalized MAE values.

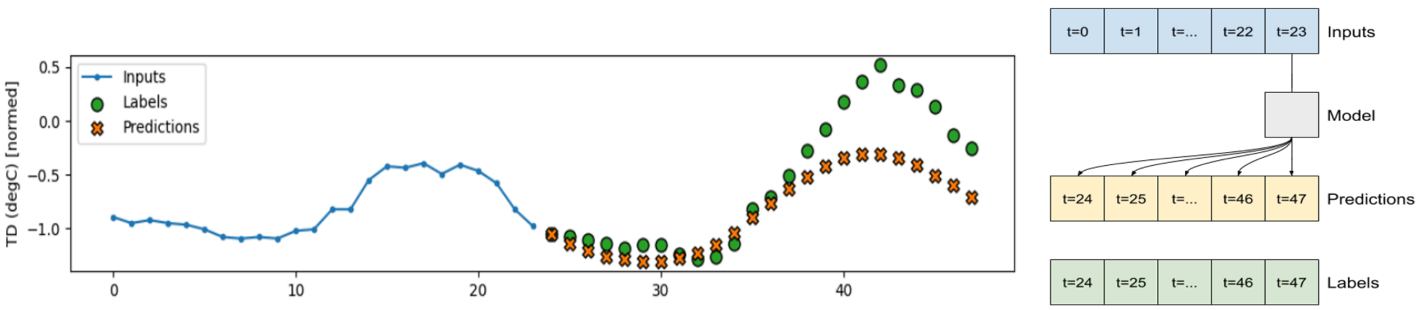


figure 21. One-Shot Linear Model Prediction.
Note. This figure demonstrates the performance of a one-shot linear model for temperature prediction (in normalized °C). The model predicts multiple time steps ahead.

Suggested Architecture & Model

This paper presents an architecture that combines *Convolutional Neural Networks (CNNs)*, *Attention Mechanisms*, and *Long Short-Term Memory networks (LSTMs)* to predict weather conditions. The model leverages the strengths of *CNNs* to capture short-term temporal patterns, attention mechanisms to model spatial dependencies between weather stations, and *LSTMs* to learn long-term temporal trends. The proposed approach aims to improve prediction accuracy by considering both local patterns and cross-station relationships.

Components Overview

1D CNNs

Recent studies have demonstrated the versatility of *1D CNNs* in various time-series forecasting tasks, beyond weather prediction. For example, *CNNs* have been used in financial markets to model stock prices, effectively capturing short-term fluctuations and trends across time. This reinforces their ability to detect fine-grained temporal patterns, which is crucial in time-series forecasting across different domains (Zeng et al., 2023). Additionally, *1D CNNs* have proven to be effective in precipitation nowcasting, where they can accurately predict short-term changes in rainfall intensity, further validating their utility in weather-related tasks (Patel et al., 2018). These successes highlight the potential of *1D CNNs* to generalize across domains and emphasize their strength in capturing local temporal dependencies.

In our Architecture, a *1D CNN* is applied to each station's data to independently learn temporal dependencies across various weather parameters. By applying *convolutional filters* across the time dimension, the model detects important local patterns, such as hourly or daily fluctuations in temperature, humidity, and other features. These short-term patterns are crucial for capturing immediate weather changes that influence predictions. After the convolutional layers extract these localized patterns, *pooling layers* are employed to reduce the dimensionality of the feature maps, ensuring that only the most significant patterns are retained for further processing.

Attention Mechanism

In addition to capturing short-term temporal patterns, our model will incorporate a *multi-head attention mechanism* to capture spatial dependencies between weather stations. The *attention mechanism* enables the model to dynamically learn how different stations influence each other by assigning attention scores based on their relevance to the target station. This approach is particularly useful when forecasting regional weather conditions, as it allows the model to weigh the importance of data from other stations in predicting the conditions of a specific station.

The incorporation of *attention mechanisms* into neural networks has demonstrated significant improvements in tasks requiring the modeling of dependencies across inputs, such as weather forecasting and machine translation (Vaswani et al., 2017). In the context of weather forecasting, *attention mechanisms* can help capture the spatial relationships between weather stations, as shown in recent studies where *attention* improves prediction accuracy by focusing on the most influential stations for each target location (Alaoui Abdellaoui & Mehrkanoon, 2021). These mechanisms allow the model to generate a refined feature map for the station being predicted, encapsulating both the temporal and spatial dependencies that affect the weather at that station.

For spatial awareness, we will use *positional encodings* based on latitude and longitude to account for geographic proximity between stations. This *positional encoding*, inspired by the approach used in *transformer models*, will help the *attention mechanism* consider the geographic relationships among stations, enabling better weather predictions based on spatial dynamics (Vaswani et al., 2017). By integrating this *attention* output with the original feature map from the pooled *CNN* layer through a *ResNet-style skip connection*, we ensure that the model retains short-term temporal features while benefiting from refined spatial information.

LSTM

To capture long-term temporal dependencies, our model uses an *LSTM layer*, which is particularly effective in handling sequential data due to its ability to retain relevant information over extended time periods while filtering out less important data via *forget gates*. This capacity to manage long-range dependencies allows the model to predict weather patterns based on longer-term trends, which is crucial for accurate forecasting (Hua et al., 2018). *LSTMs* have proven successful in various time-series prediction tasks, including financial forecasting, where they capture both short-term fluctuations and long-term trends in highly dynamic environments (Livieris et al., 2020).

In our architecture, the input data will be processed through the *CNN* and *attention layers*, after which a *ResNet-style skip connection* will combine the original pooled CNN output with the attention-refined output for the target station. This combined output will be fed into the *LSTM*, enabling it to learn from both the short-term patterns captured by the *CNN* and the spatial dependencies captured by the attention mechanism. The *skip connection* ensures that no important short-term temporal features are lost, while the *LSTM* processes this combined output to capture the long-term dependencies needed for accurate weather prediction.

Proposed Architecture

The proposed model will combine *1D Convolutional Neural Networks (CNNs)*, a *multi-head attention mechanism*, *ResNet-style skip connections*, and *Long Short-Term Memory Networks (LSTMs)* to address both short-term and long-term (window size) weather forecasting. The architecture is designed to capture short-term temporal patterns, spatial dependencies between weather stations, and long-term trends.

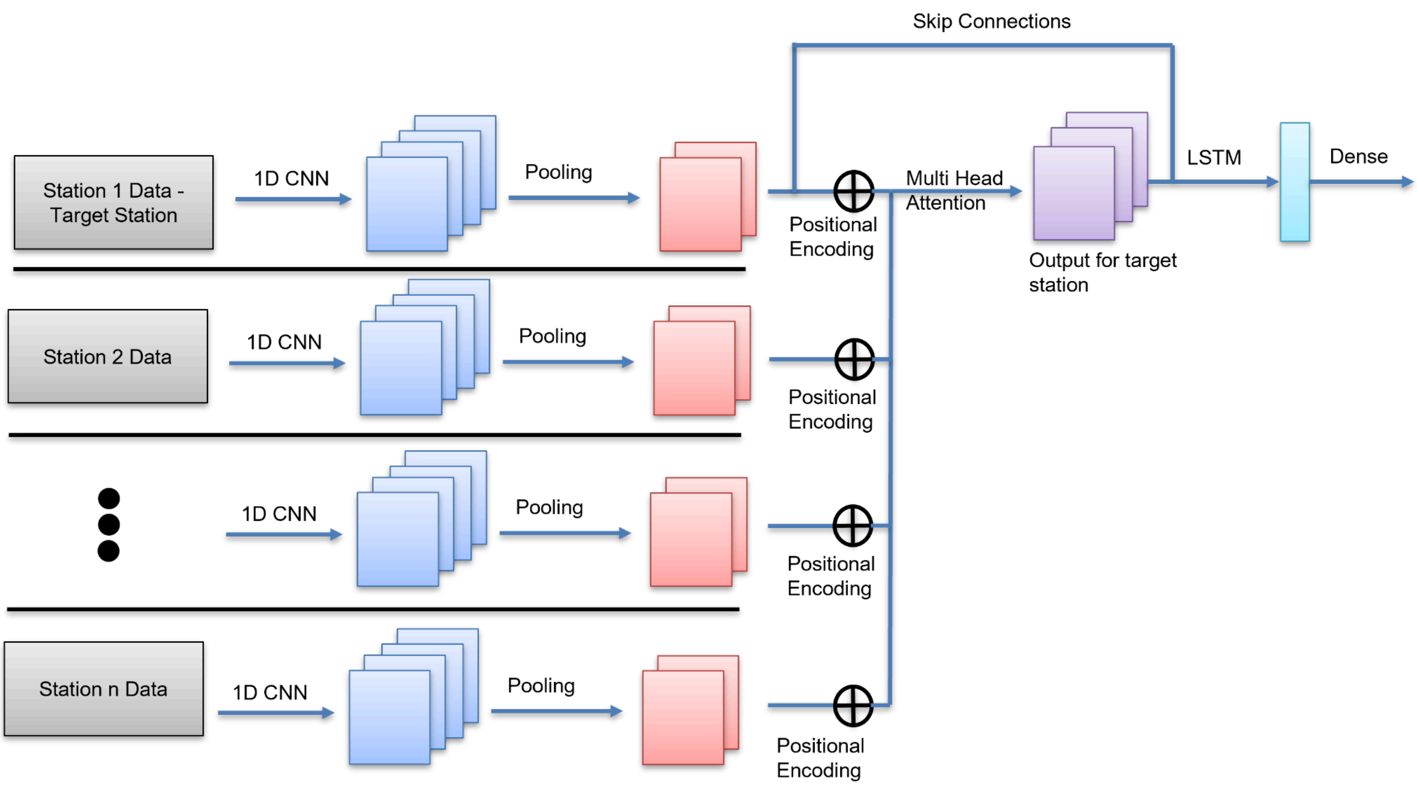


figure 22. *The Proposed Architecture.*

Architecture Overview

The architecture will be structured to forecast weather by learning both temporal dependencies (short-term and long-term) and spatial dependencies (between stations). Below are the main components:

1. *1D CNN* is first applied to each station's data to capture short-term temporal patterns in weather features like temperature and wind speed.
2. *Pooling* is used to reduce the dimensionality of the feature maps while retaining the most important information.
3. *Multi-head attention* is applied across stations to refine the feature map for the target station by incorporating information from neighboring stations. This allows the model to learn spatial dependencies, ensuring that the prediction for one station benefits from data observed at others.
   The output will be a spatially-refined feature map for the target station, representing how the weather at the target station is influenced by other stations.
4. A *ResNet-style skip connection* combines the original pooled output (containing short-term temporal features) and the attention output (containing refined spatial dependencies). This will ensure that the original short-term patterns from the target station are preserved.
5. The combined feature map will be passed into an *LSTM layer*, which will learn long-term temporal dependencies from the target station's weather data.
   The *LSTM* will process the time-series data to predict weather trends over the forecast horizon.
6. Finally, a *fully connected layer* is applied to the *LSTM's* output to generate the weather prediction for the target station at the next time step. This could be the next hour's forecast (for recursive prediction) or a multi-day forecast (for one-shot prediction).

This architecture allows the model to process both short-term temporal data and long-term trends while incorporating spatial relationships between stations.

Ensemble Methods: Short-Term and Long-Term: The model will support two forecasting strategies: *recursive forecasting* for short-term predictions and *one-shot forecasting* for long-term predictions. The exact time horizons for both short-term and long-term forecasting will need to be determined during hyperparameter tuning.

Training Overview

In this proposed architecture, two models will be developed for each station for which we aim to predict weather conditions. Both models will share the same structure, but they will differ in their output layers to accommodate different forecasting strategies:

- The first model will be designed for *recursive forecasting*, predicting one hour ahead. This model will iteratively use each predicted time step as input for the next, ensuring a step-by-step forecast for short-term weather changes.
- The second model will support the *one-shot forecasting* approach, where the entire sequence of future predictions will be made in a single forward pass. This model will focus on long-term forecasts over several days.

Each of these models will be trained independently for each station, as the spatial and temporal dynamics of weather may differ between locations. In our case, we aim to predict the temperature for three stations, meaning that separate models will be trained for each station. As a result, a total of six models (two models per station) will be developed - one for short-term recursive forecasting and one for long-term one-shot forecasting for each station.

This approach will ensure that the model is tailored to the unique weather patterns of each station, allowing for more accurate predictions in both short-term and long-term horizons.

Hyperparameters

What Hyperparameter is

Hyperparameters are key components in machine learning models that define how a model is structured and trained. hyperparameters must be set before training begins and can significantly impact model performance. The process of tuning these hyperparameters is crucial for optimizing the model's ability to generalize well on unseen data.

The Hyperparameters in Our Architecture

In our architecture, several hyperparameters need to be carefully examined and tuned during implementation to ensure optimal performance. The following hyperparameters will be considered:

1. ***Learning Rate***: The learning rate controls how much the model's weights are updated with respect to the gradient during backpropagation. A learning rate that is too high can cause the model to converge too quickly to a suboptimal solution, while a learning rate that is too low can result in slow convergence.
2. ***Batch Size***: The batch size determines how many data samples are processed before the model updates its parameters. Larger batch sizes lead to more stable gradients, while smaller batch sizes introduce more noise but might help the model escape local minima.
3. ***Number of Filters (CNN)***: In the *CNN layer*, the number of filters determines how many patterns the model will learn to extract from the data. More filters can allow the model to capture more complex features, but too many filters may lead to overfitting.
4. ***Kernel Size (CNN)***: The kernel size in the *1D CNN* controls the width of the convolution window across the time dimension. A larger kernel can capture broader patterns, while a smaller kernel will focus on fine-grained local patterns.
5. ***Hidden Units (LSTM)***: The number of hidden units in the *LSTM layer* affects the model's ability to capture long-term dependencies. Too few units might limit the model's capacity to understand complex temporal patterns, while too many units can lead to overfitting.
6. ***Attention Heads***: In the attention mechanism, the number of heads controls how many independent attention layers are learning spatial dependencies. More attention heads may help the model capture more complex relationships between stations, but increasing the number of heads also increases computational cost.
7. ***Recursive vs. One-Shot Forecasting Horizon***: In the recursive approach, the model predicts one step ahead, while in the one-shot approach, it predicts the entire sequence at once. The time horizons for both approaches will need to be tuned.
8. ***Time Window Size***: The time window size parameter refers to the input-output time frames used for forecasting. We will experiment with different input window sizes (e.g., 24 hours, 48 hours) and corresponding output windows (e.g., 1 hour, 12 hours, 24 hours). This will help determine the most effective window size for capturing both short-term fluctuations and longer-term trends in weather patterns.
9. ***Epochs***: The number of epochs defines how many times the entire training dataset is passed through the model. Too few epochs may result in underfitting, where the model has not learned enough from the data. Too many epochs may lead to overfitting, where the model learns patterns specific to the training data and fails to generalize well to unseen data.

We will systematically examine different hyperparameters, including learning rate, batch size, CNN filters, kernel size, LSTM hidden units, attention heads, time window size, Epochs and forecasting horizons to optimize the model's performance and ensure accurate weather predictions.

Cyclic Parameters

In weather forecasting, many natural phenomena exhibit cyclic patterns that repeat over regular intervals. These patterns, such as the daily cycle of sunrise and sunset, or the seasonal transitions throughout the year, play a critical role in shaping the behavior of weather conditions. For example, temperature, humidity, and wind direction often follow predictable cycles based on the time of day or season. Capturing these cyclical behaviors is essential for creating an accurate forecasting model.

Time

In our proposed architecture, time will be treated as a fundamental cyclic parameter, crucial for capturing natural cycles in weather data. Time-driven patterns such as daily temperature fluctuations and seasonal changes are key elements of weather forecasting. For example, the transition from 00:00 to 23:00 is part of a continuous cycle, but if treated as linear, these times may seem far apart. Instead, we want our model to understand the cyclical flow of time and its role in recurring weather patterns.

We will achieve this by transforming time into sine and cosine values, allowing the model to recognize that time repeats in cycles. This transformation helps the model to understand that time itself is not a linear progression, but a continuous loop that affects weather conditions regularly.

<u>Hour of the Day:</u>

$$DaySin = sin(timestamp * (\frac{2\pi}{day}))$$

$$DayCos = cos(timestamp * (\frac{2\pi}{day}))$$

By encoding the day of the year cyclically, the model will gain an understanding of the yearly cycle, recognizing the progression through different seasons - such as the shift from winter to spring or summer to fall - enabling it to more accurately capture seasonal weather patterns and changes.

Day of the Year:

$YearSin = sin(timestamp * (\frac{2\pi}{year}))$

$YearCos = cos(timestamp * (\frac{2\pi}{year}))$

Wind

Wind patterns are another critical aspect of weather forecasting, and just like time, wind direction exhibits cyclical behavior. Wind direction is typically measured in degrees, with values ranging from 0° to 359°. However, treating these degrees as linear numbers introduces a problem: 0° and 359° are very close in terms of wind direction (both represent nearly the same direction), but a model that treats them linearly might perceive them as far apart.

To address this, we will convert wind direction into a vector representation. This way, the model understands that 359° and 0° are essentially the same direction, ensuring that it accurately interprets changes in wind direction.

Here's the plan for how we will handle wind data:

Wind Speed (wv) and Max Wind Speed (max_wv):

Wind speeds will remain in their original form but will be combined with wind direction components to ensure that directional information is captured accurately.

Wind Direction (wd):

Wind direction, originally given in degrees (0° to 359°), will be converted to radians. This allows us to apply trigonometric functions and break the wind direction into two components:

1. $W_x = wv \times cos(wd_{rad})$
2. $W_y = wv \times sin(wd_{rad})$

Max Wind Speed Components:

Similarly, the maximum wind speed will be decomposed into x and y components using the same trigonometric approach.

Forecast Evaluation & Verification

The accuracy and reliability of the suggested architecture are essential for successful mid-term weather prediction in Israel. This section outlines the strategies and metrics used to evaluate and verify the model's performance, ensuring it meets project goals.

Evaluation Metrics

To evaluate the performance of the WeatherNet forecasting model, we will use the following metrics:

- *Mean Absolute Error (MAE)*: This metric measures the average magnitude of the errors in predictions, without considering their direction. It is useful for understanding the overall precision of the model's predictions compared to the actual weather data.
- *Root Mean Squared Error (RMSE)*: *RMSE* provides insight into the model's error rate by penalizing larger errors more heavily. This metric is particularly useful for ensuring that large deviations between predicted and actual values are minimized, which is crucial for accurate weather forecasting.
- *R^2 Score (Coefficient of Determination)*: The R^2 score measures the proportion of the variance in the temperature data that is predictable by the model. An R^2 score close to 1 indicates that the model effectively captures variability, making it a key measure of the model's predictive power.

Cross-Validation

To ensure that the model generalizes well across different weather conditions and time periods, we will implement *K-Fold Cross-Validation*. This technique divides the dataset into *K* subsets and iteratively trains the model on *K-1* subsets while validating it on the remaining subset. By repeating this process across multiple folds, we can assess the model's robustness and reduce the risk of overfitting to specific segments of the data.

Model Sanity Check

In addition to evaluating the machine learning model, we will compare its performance to baseline models such as linear regression and persistence models, which predict future weather conditions based on the assumption that they will remain unchanged. This comparison helps ensure that the complexity of the machine learning model is justified and that it outperforms traditional, simpler forecasting methods.

Comparison with IMS Forecasts

To further validate the accuracy of the machine learning model, we will compare its forecasts with traditional weather predictions provided by the *Israel Meteorological Service (IMS)*. This comparison will help determine how well the machine learning model performs against established meteorological methods. Achieving comparable or superior accuracy to IMS will be considered a key success criterion for the WeatherNet System.

Future Development Roadmap

After completing the project's first phase, which involved comprehensive research and exploring potential architectural solutions, the next stage focuses on the practical implementation of the system. This phase translates the theoretical findings into a functional solution, including the development of machine learning models, data integration, and the creation of a website to deliver accurate weather predictions and system statistics.

Data Collection and Integration

The data collection phase involves retrieving real-time weather data from the *Israel Meteorological Service (IMS)* through their API. This data includes essential weather parameters such as temperature, humidity, wind speed, and precipitation, gathered from multiple stations across Israel.

Data Preprocessing

After collection, the data undergoes preprocessing to address inconsistencies, handle missing values, and prepare it for model training:

- *Missing Data Handling*: Missing values are imputed using techniques such as interpolation or filling based on data from nearby stations.
- *Cyclic Data Processing*: Cyclic variables, like time of day and seasonality, are transformed into representations that machine learning models can interpret, ensuring the capture of cyclic weather patterns.
- *Normalization*: Weather parameters are normalized to ensure that all data is on a comparable scale, improving the performance of the machine learning algorithms.

Time Series Preparation

The preprocessed data is converted into time series format by creating sliding windows of past data points, which allows the models to learn temporal dependencies in weather patterns. Various window sizes are tested during this phase to find the optimal configuration for accurate predictions.

Model Development and Training

The model utilizes a hybrid architecture combining *CNNs*, *LSTMs*, and an *Attention mechanism* to capture both short-term and long-term weather patterns while accounting for spatial dependencies between weather stations. The *CNNs* handle short-term temporal patterns, while the attention mechanism models spatial relationships between stations. The combined output is passed to an *LSTM* layer for long-term trend forecasting.

Two approaches are used: *recursive forecasting* for short-term predictions and *one-shot forecasting* for long-term predictions. Key hyperparameters will be tuned to optimize model accuracy.

Evaluation & Verification

Once the model is trained, it is tested using cross-validation to ensure generalizability. Performance is measured using key metrics such as *Mean Absolute Error (MAE)* and *Root Mean Squared Error (RMSE)*. This phase also includes a comparison of the machine learning model's forecasts with traditional IMS forecasts to evaluate the improvement in accuracy. For more detailed information on the Evaluation procedures and Verification methods, please refer to the *Forecast Evaluation & Verification* section.

Testing

In future development, the system as a whole will be thoroughly tested using the test cases defined in the *System Testing* section. These test cases will cover all aspects of the system, including data processing, model performance, and the integration of various components. The goal is to ensure that the entire pipeline—from data ingestion to forecast generation—functions correctly, reliably, and efficiently under a wide range of conditions, including edge cases and extreme weather events.

Website UI Development

This phase focuses on designing a user-friendly interface that allows users to easily interact with the WeatherNet system and access accurate weather forecasts and system statistics. The website's structure will align with the outline provided in the *System Goals* section of this paper.

Deployment

The deployment phase involves integrating the trained model with the website and hosting the application on a cloud platform (e.g., AWS or GCP). The system is designed to provide real-time updates as new weather data becomes available through the *IMS* API. The deployment architecture ensures scalability and efficient handling of multiple user requests, ensuring that forecasts are delivered quickly and accurately.

Anticipated Challenges

Developing an accurate machine-learning-based weather forecasting system involves several technical and operational challenges. As we progress through the different stages of this project, we anticipate encountering the following issues:

Data Availability, Quality, and Missing Data

One of the primary challenges is ensuring access to high-quality and consistent weather data. While the Israel Meteorological Service (IMS) provides comprehensive data from numerous weather stations, missing or corrupted data can still be a significant issue due to station malfunctions, communication errors, or data transmission delays. Missing data disrupts the temporal patterns crucial for accurate time-series forecasting, leading to potential degradation of model performance. To address this, we plan to implement robust data preprocessing techniques, including missing data imputation methods such as interpolation and leveraging data from nearby stations.

Real-Time Forecasting Integration

Integrating real-time data from the *IMS* API presents another challenge, as delays or interruptions in data transmission can affect the accuracy and timeliness of the forecasts. Real-time data processing must be reliable and efficient to ensure that forecasts are based on the most recent information. We may implement monitoring systems to detect data transmission issues and develop fallback strategies using past data to fill temporary gaps.

High Computational Costs for Model Training

Training complex models like *CNNs* and *LSTMs* on large weather datasets demands significant computational power, leading to long training times and high hardware resource consumption. This challenge is compounded by the need for extensive hyperparameter tuning. To address this, we will explore distributed computing and cloud-based GPU environments, though these solutions may increase operational costs. Efficient resource management will be key to balancing accuracy and training time.

Hyperparameter Tuning

Fine-tuning hyperparameters such as *learning rates*, *batch sizes*, and the *number of layers in CNNs and LSTMs* can contribute to improving the performance of our forecasting models. The challenge lies in finding the optimal values for these parameters to balance model performance and training time. Given the complexity of our model architecture, hyperparameter tuning is computationally expensive and time-consuming. To manage this, we will employ grid search and random search techniques to streamline the tuning process.

Climate Change and Long-Term Trends

Climate change introduces shifting baselines and trends in weather data, which could affect the accuracy of long-term predictions. The model must be adaptable to evolving patterns while maintaining accuracy in the short and mid-term. To address this, we will consider retraining our models with updated data and introducing features that capture long-term trends.

Tools and Components We Will Use

To develop an accurate, efficient, and approachable machine-learning-based weather forecasting system, we will utilize a variety of tools and components. These technologies cover everything from data manipulation and model development to real-time data integration and user-friendly web deployment.
Below is a list of the key technologies and libraries we plan to use:

Python – The primary programming language for developing machine learning models and handling data processing.

Pandas Library – Used for data manipulation and analysis, particularly for working with large weather datasets.

NumPy Library – Essential for performing numerical operations and handling array-based data efficiently.

TensorFlow/Keras Library – Deep learning frameworks to build, train, and deploy neural networks for weather prediction.

Matplotlib Library – A key library for creating visualizations of data trends and model outputs.

Seaborn Library – Built on top of *Matplotlib*, it allows for easy and aesthetically pleasing statistical data visualizations.

Jupyter Notebooks – Provides an interactive environment for developing, testing, and sharing code and models.

IMS API – The *Israel Meteorological Service* API for integrating real-time weather data into the system.

Cloud-based GPUs – Used for scaling model training on large datasets, offering faster processing with distributed computing.

Git – A version control system for managing and tracking code changes across team members.

React – The frontend framework used to build an approachable and dynamic user interface for the forecasting website, including *HTML*, *JavaScript*, and *CSS* for interactive and responsive design.

Ant Design – A design library to ensure a consistent and intuitive user interface on the web platform.

Cloud storage solution – Used for storing the website and any real-time data, such as *AWS S3* or *Google Cloud Storage*.

Missing Data Imputation Libraries – Libraries like *Scikit-learn*, *FancyImpute*, *Datawig*, and *MissForest* to handle missing weather data efficiently.

Product

Requirements

The WeatherNet project aims to develop a machine-learning system for accurate temperature prediction using data from the *Israeli-Meteo Service (IMS)*. The system will provide forecasts via a website interface, offering users temperature predictions, model performance reports, and visualizations.

Below are the system's functional and non-functional requirements.

Functional Requirements

ID	Requirement	Description
FR-1	Data Collection	The system shall collect data from the *Israeli-Meteo Service (IMS)* API of numerous station recordings, ensuring continuous data flow for training and forecasting purposes.
FR-2	Data Preprocessing	The system shall clean and preprocess the collected data, including handling missing values through imputation and transforming cyclic parameters into numerical formats for model input.
FR-3	Time Series Preparation	The system shall convert raw data into time series windows suitable for input into the forecasting models, following the guidelines specified in the *TimeSeries Analysis* section of the paper.
FR-4	Temperature Forecasting	The system shall generate temperature forecasts at various time intervals using our defined ML architecture.
FR-5	Multi-Step Forecasting	The system shall support multi-step temperature forecasting using recursive or one-shot prediction approaches to provide accurate predictions over extended future time periods.
FR-6	Model Training and Tuning	The system shall enable model training on historical temperature data and support hyperparameter tuning to optimize key parameters such as learning rates, batch sizes, and model architecture.
FR-7	Error Metrics Calculation	The system shall compute error metrics, including *Mean Absolute Error (MAE)* and *Root Mean Squared Error (RMSE)*, to evaluate the accuracy of temperature forecasts.
FR-8	Forecast Visualization	The system shall provide visualizations (e.g., line plots, bar charts) comparing temperature forecast outputs to actual past data.
FR-9	Website Interface	The system shall provide a web-based user interface (UI) where users can access the trained system.

ID	Requirement	Description
FR-10	Reporting	The system shall automatically generate reports that summarize temperature forecast accuracy, error metrics, and model performance over specified time periods.
FR-11	Real-Time Data Update	The system shall update forecasts in real-time as new data is ingested, ensuring predictions reflect the most recent temperature readings.
FR-12	Community Sharing	The system shall allow users to share temperature forecasts, visualizations, and model performance reports via built-in sharing features in the website.
FR-13	User Support System	The system shall provide a user support feature within the website, including access to a help center, FAQ section, and direct contact options (e.g., email).

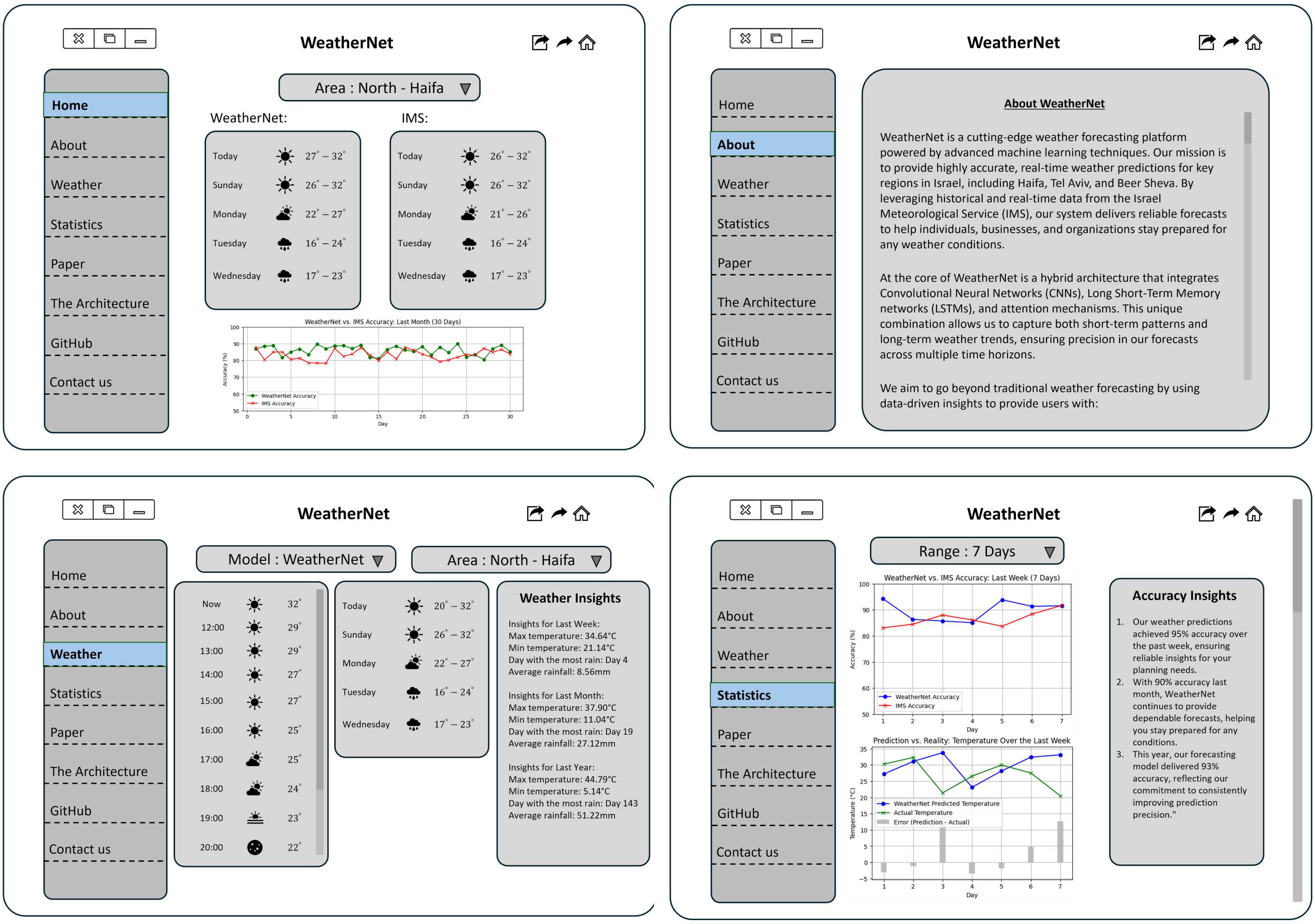
Non-Functional Requirements

ID	Requirement	Description
NFR-1	Performance	The system shall generate temperature forecasts within a maximum response time of 20 seconds per forecast request.
NFR-2	Scalability	The system shall be scalable to handle large datasets from multiple weather stations and support future extensions (e.g., additional locations, higher resolution data).
NFR-3	Accuracy	The system shall achieve a forecast accuracy with a *Root Mean Squared Error (RMSE)* of less than 1.5°C for temperature predictions, as specified in the *Forecast Precision Expectations* section of the paper.
NFR-4	Maintainability	The system shall be designed with modularity and well-documented code to support easy maintenance, updates, and the integration of new models or data sources without requiring significant architectural changes.
NFR-5	Usability	The web-based interface shall be user-friendly and intuitive, providing users with clear temperature predictions, visualizations, and performance insights without requiring deep technical knowledge.
NFR-6	Resource Efficiency	The system shall optimize resource usage (e.g., CPU, memory, and storage) for both training and inference tasks, ensuring efficient operation on both local hardware and cloud platforms.
NFR-7	Extensibility	The system shall be designed to allow the addition of new forecasting methods or model improvements with minimal refactoring, enabling future research and model extensions.
NFR-8	Cloud Deployment	The system shall be capable of being deployed on a cloud platform (e.g., AWS, GCP) to ensure scalability and availability for users regardless of geographic location.

WeatherNet Website Diagrams

The diagram below provides an overview of the WeatherNet platform's user interface, which serves as the main access point for real-time weather predictions. It includes sections for weather forecasts, model accuracy insights, and an interactive comparison tool that highlights the prediction precision.

This web-based platform is designed to be intuitive and informative, ensuring that users can easily explore forecast data and the machine learning methods behind it.



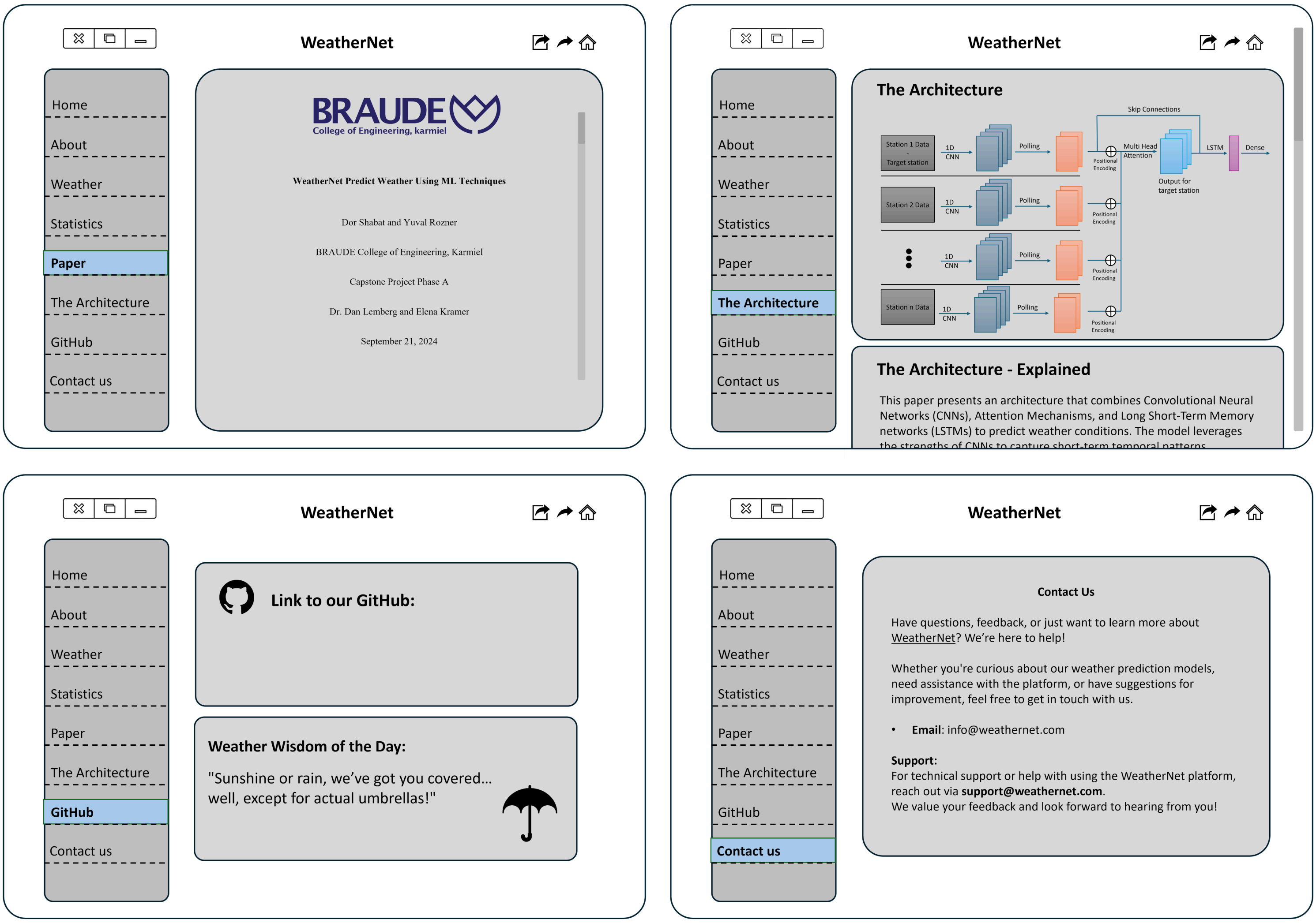


figure 23. WeatherNet Web Application Screens Overview.

Activity Diagram of the System

The following activity diagram illustrates the key operational flows within the WeatherNet system. It covers three primary use cases:

1. The system manager trains the forecasting model.
2. The user requests and receives weather forecasts through the web interface.
3. The user accesses forecast accuracy statistics via the website.

Each flow outlines the interaction between the primary system components.

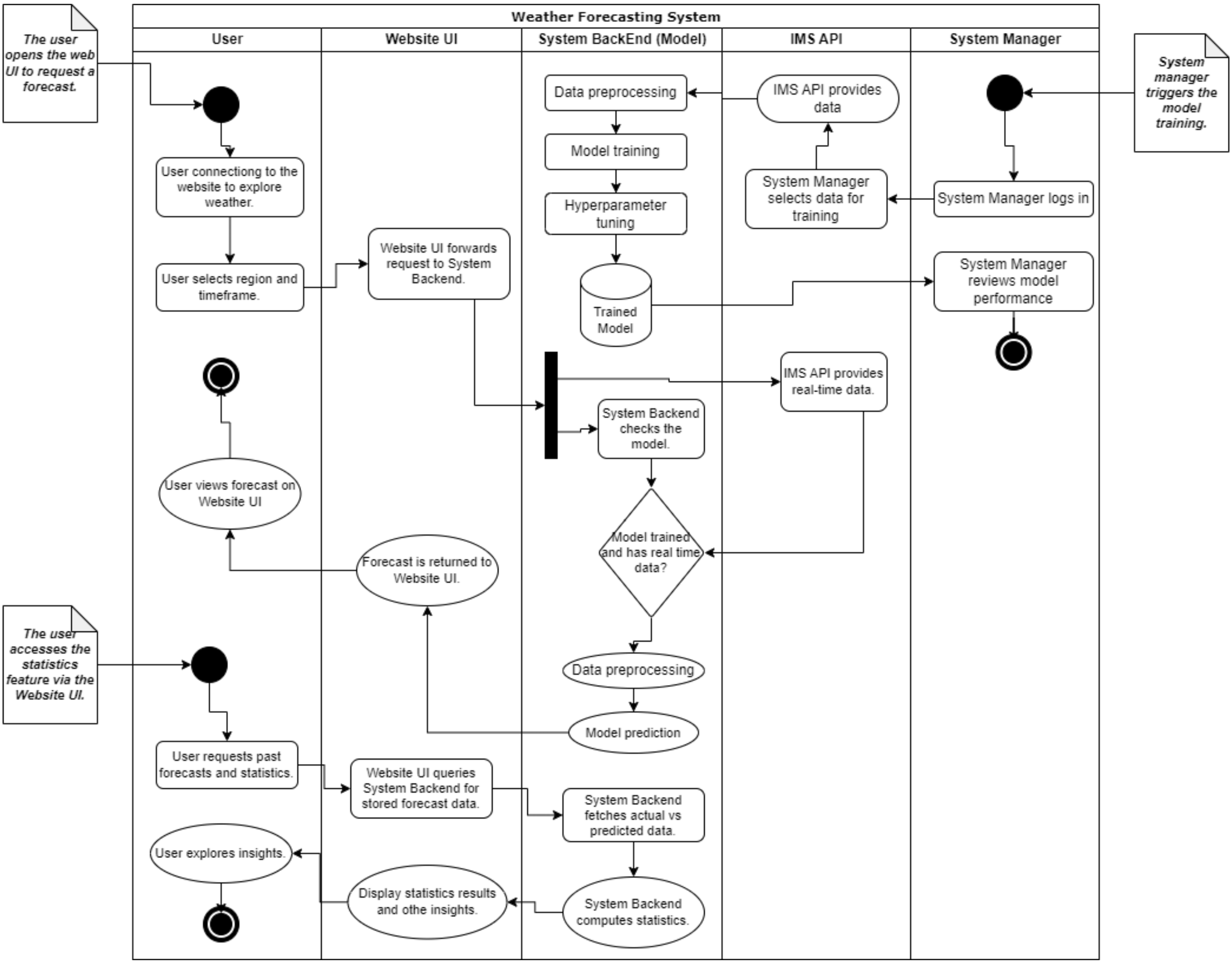


Figure 24. *System Activity Diagram. [made by Draw.io].*

System Testing

System testing is a critical phase designed to verify the accuracy, reliability, and performance of both the weather prediction model and the UI website component. By simulating real-world conditions, we aim to ensure that each part of the system—from data ingestion and processing to forecast generation and user interaction—functions as expected. Below, we outline the primary tests that will be conducted to validate the system's performance.

#	Test	Expected Outcome	Testing Approach
1	Unit Testing (Data Preprocessing)	Handle missing/invalid data correctly. Clean dataset output.	Use analysis functions to validate data output and preprocessing.
2	Unit Testing (Model Training)	Model trains without errors, convergence in loss metrics.	Train on small datasets, verify correct output & loss decrease.
3	Integration Testing (Data Pipeline)	Smooth data flow from ingestion by the API to preprocessing.	Test full pipeline on sample data, and check for consistent output.
4	Performance Testing (Model)	The model achieves expected MAE, RMSE on test data.	Evaluate the test set, and compare metrics with expected values.
5	Functionality Testing (UI)	UI components respond correctly to user actions.	Manually test UI interactions or use Selenium for automation.
6	User Satisfaction Testing (UX)	UI and forecasts meet user expectations in terms of usability and accessibility.	Conduct usability tests with real users, gather feedback on interaction flow, and forecast readability.
7	UI Responsiveness Testing	UI adapts to various screen sizes and devices (Mobile & Desktop).	Test UI on multiple devices, ensuring proper display and functionality.
8	Load Testing (System)	The system remains stable under high data loads.	Simulate heavy data loads and monitor system performance.
9	Real-Time Testing (Data Ingestion)	Real-time data is processed without delays or errors.	Feed live data streams, and check for correct and timely ingestion.
10	Real-Time Testing (Forecast Accuracy)	Forecasts stay accurate with real-time data.	Compare real-time predictions with ground truth weather data.

11	End-to-End Testing	The entire system works from data ingestion to forecast output.	Simulate real scenarios from data input to forecast delivery.
12	Scalability Testing	The system remains performant as the number of users or data volume increases.	Simulate increased user load and larger datasets, monitor performance metrics (latency, throughput).
13	Offline Functionality Testing	Verify that the system can continue processing data and forecasts if disconnected from certain external services.	Simulate the loss of external services (e.g., data providers), and ensure the system functions with cached data or default behavior.

References

Ahmadi, A., Zahra Zargaran, Azadeh Mohebi, & Taghavi, F. (2014). *Hybrid model for weather forecasting using ensemble of neural networks and mutual information.* https://doi.org/10.1109/igarss.2014.6947305

Abdellaoui, I., & Mehrkanoon, S. (2021). *Deep multi-stations weather forecasting: explainable recurrent convolutional neural networks.* https://arxiv.org/pdf/2009.11239v6.pdf

Bochenek, B., & Ustrnul, Z. (2022). *Machine Learning in Weather Prediction and Climate Analyses—Applications and Perspectives*. Atmosphere, 13(2), 180. https://doi.org/10.3390/atmos13020180

Casolaro, A., Capone, V., Iannuzzo, G., & Camastra, F. (2023). *Deep Learning for Time Series Forecasting: Advances and Open Problems*. Information, 14(11), 598. https://doi.org/10.3390/info14110598

European Centre for Medium-Range Weather Forecasts. (n.d.). *About ECMWF. European Centre for Medium-Range Weather Forecasts*. https://www.ecmwf.int/en/about

European Centre for Medium-Range Weather Forecasts. (n.d.). *Integrated Forecasting System. ECMWF.* https://www.ecmwf.int/en/forecasts/documentation-and-support/changes-ecmwf-model

Hua, Y., Zhao, Z., Li, R., Chen, X., Liu, Z., & Zhang, H. (2018). *Deep Learning with Long Short-Term Memory for Time Series Prediction.* https://arxiv.org/pdf/1810.10161v1.pdf

Israel Meteorological Service. (n.d.). *Israel Meteorological Service.* https://ims.gov.il/en

Israel Meteorological Service. (n.d.). *About Israel Meteorological Service.* https://ims.gov.il/en/about

Israel Meteorological Service. (n.d.). *COSMO model*. https://ims.gov.il/en/COSMO

Israel Meteorological Service. (n.d.). *Forecast tools*. https://ims.gov.il/en/ForecastTools

Israel Meteorological Service. (n.d.). *Observation data API.* https://ims.gov.il/en/ObservationDataAPI

Israel Meteorological Service. (n.d.). *Meteorological stations.* https://ims.gov.il/en/stations

Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). *A CNN–LSTM model for gold price time-series forecasting. Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-04867-x

Max Planck Institute for Meteorology. (n.d.). *Weather and Climate Model ICON.* https://www.mpimet.mpg.de

Met Office. (n.d.). *Weather forecasting using the Met Office unified model. Met Office.* https://www.metoffice.gov.uk/research/approach/modelling-systems/unified-model/weather-forecasting

National Centers for Environmental Prediction. (n.d.). *Global forecast system (GFS). U.S. National Weather Service.* https://www.emc.ncep.noaa.gov/emc/pages/numerical_forecast_systems/gfs.php

Patel, M., Patel, A., & Ghosh, R. (n.d.). *Precipitation Nowcasting: Leveraging bidirectional LSTM and 1D CNN* https://arxiv.org/pdf/1810.10485.pdf

Temperature. (2023, September 6). *Wikipedia*. https://en.wikipedia.org/wiki/Temperature

TensorFlow. (n.d.). *Sliding window approach.* https://raw.githubusercontent.com/tensorflow/docs/master/site/en/tutorials/structured_data/images/split_window.png.

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017). *Attention Is All You Need.* https://arxiv.org/pdf/1706.03762v7.pdf

Weather. (2023, August 28). *Wikipedia*. https://en.wikipedia.org/wiki/Weather

Wind. (2023, August 30). *Wikipedia*. https://en.wikipedia.org/wiki/Wind

Zeng, Z., Kaur, R., Siddagangappa, S., Rahimi, S., Balch, T., Veloso, M., & Morgan, J. (2023). *Financial Time Series Forecasting using CNN and Transformer.* https://arxiv.org/pdf/2304.04912.pdf